

PEARSON

移动开发经典丛书



iCloud开发秘籍

——实践掌握iOS & OS X应用的数据管理技术

Learning iCloud Data Management:
A Hands-On Guide to Structuring Data for iOS and OS X

[美] Jesse Feiler 著
张杰 赵晓欣 译

PEARSON

清华大学出版社

移动开发经典丛书

iCloud 开发秘籍—— 实践掌握 iOS & OS X 应用的数据管理技术

[美] Jesse Feiler 著

张 杰 赵晓欣 译

清华大学出版社

北 京

Jesse Feiler

Learning iCloud Data Management: A Hands-On Guide to Structuring Data for iOS and OS X

EISBN: 978-0-321-88911-9

Authorized translation from the English language edition, entitled Learning iCloud Data Management: A Hands-On Guide to Structuring Data for iOS and OS X, 978-0-321-88911-9 by Jesse Feiler, published by Pearson Education, Inc, publishing as Addison-Wesley, Copyright © 2014.

All rights reserved. No part of this book may be reproduced or transmitted in any form or by any means, electronic or mechanical, including photocopying, recording or by any information storage retrieval system, without permission from Pearson Education, Inc.

CHINESE SIMPLIFIED language edition published by PEARSON EDUCATION ASIA LTD., and TSINGHUA UNIVERSITY PRESS Copyright © 2015.

北京市版权局著作权合同登记号 图字: 01-2014-3247

本书封面贴有 Pearson Education(培生教育出版集团)防伪标签, 无标签者不得销售。

版权所有, 侵权必究。侵权举报电话: 010-62782989 13701121933

图书在版编目(CIP)数据

iCloud 开发秘籍——实践掌握 iOS & OS X 应用的数据管理技术/(美) 菲勒尔(Feiler, J.) 著;
张杰, 赵晓欣 译. —北京: 清华大学出版社, 2015

(移动开发经典丛书)

书名原文: Learning iCloud Data Management: A Hands-On Guide to Structuring Data for iOS and OS X
ISBN 978-7-302-38720-6

I. ①i… II. ①菲… ②张… ③赵… III. ①移动终端—应用程序—程序设计 IV. ①TN929.53

中国版本图书馆 CIP 数据核字(2014)第 284035 号

责任编辑: 王 军 李维杰

封面设计: 牛艳敏

版式设计: 思创景点

责任校对: 邱晓玉

责任印制: 何 芊

出版发行: 清华大学出版社

网 址: <http://www.tup.com.cn>, <http://www.wqbook.com>

地 址: 北京清华大学学研大厦 A 座 邮 编: 100084

社 总 机: 010-62770175

投稿与读者服务: 010-62776969, c-service@tup.tsinghua.edu.cn

质 量 反 馈: 010-62772015, zhiliang@tup.tsinghua.edu.cn

印 刷 者: 三河市君旺印务有限公司

装 订 者: 三河市新茂装订有限公司

经 销: 全国新华书店

开 本: 185mm×260mm

印 张: 20

字 数: 499 千字

版 次: 2015 年 1 月第 1 版

印 次: 2015 年 1 月第 1 次印刷

印 数: 1~3500

定 价: 59.80 元

产品编号: 059578-01

译者序

计算机领域每天都在高速发展,新语言和新技术的产生速度早已超过了一般人的学习速度。对于计算机或相关领域的从业人员或兴趣爱好者而言,学习是为了更好地应用。在这种情况下,有选择地学习最有用的知识,以及甄别各种计算机知识的重要程度和相互关系的能力就显得非常重要。根据网络分析公司Net Applications的数据,截至2014年8月,OS X的市场占有率已经达到6.38%,而且仍在稳步增长。OS X不仅占据了各种高端专业领域,在服务器领域中也能见到其身影。在移动平台上,iOS(包括手机和平板电脑)的市场占有率达到了43.86%。各种数据表明,苹果系的操作系统——OS X和iOS——正变得越来越重要。而随着应用程序快速进入企业和云平台,iOS和OS X平台的开发人员需要学习新的数据管理技术。而现在这本书正是一本帮助开发者了解数据管理技术的良好指南,它很好地填充了开发人员的学习需求。

《iCloud开发秘籍——实践掌握iOS & OS应用的数据管理技术》是一本介绍iCloud数据管理的图书。作者是Apple的数据库专家Jesse Feiler,在本书中,他对如何通过Apple最新的API和技术管理和同步各种格式的数据进行了详细的阐述和介绍。同时,本书会得出有效的解决方案,并且提供iOS和OS X间无缝同步高度可用的应用程序。另外,本书将会引导读者完成Apple的几个关键数据管理技术的整合,包括地址簿和日历API。作者同时也对如何管理数据进行了详细介绍,读者通过阅读本书可以了解到如何创建高质量的Cocoa和Cocoa Touch用户界面,以及如何进行快速可靠的iCloud同步过程,也会学习到如何将自己的用户数据与Apple的标准应用程序数据结构结合在一起,并且创建功能丰富可同步性高的环境。本书适合于有Objective-C开发基础和Xcode使用经验,以及对数据库设计有基本认知的读者阅读。

在这里要感谢清华大学出版社不遗余力地引进国外优秀IT类书籍,感谢编辑们在本书翻译阶段给予我们的鼓励与支持,没有她们,本书不可能顺利付梓。

本书全部章节由张杰、赵晓欣翻译,参与本次翻译活动的还有孔祥亮、陈跃华、杜思明、熊晓磊、曹汉鸣、陶晓云、王通、方峻、李小凤、曹晓松、蒋晓冬、邱培强、洪妍、李亮辉、高娟妮、曹小震、陈笑。在此一并表示感谢。

翻译本书的过程中,译者虽已尽最大努力确保专业术语的统一和准确,也尽最大努力将原作者的意境用朴实的中文展现给读者,但囿于个人的水平,书中的问题和疏漏之处在所难免,敬请读者朋友给予批评指正。

总之,希望读者能够通过阅读本书理解到如何使用OS X和iOS平台中的iCloud API,并顺利了解如何对iCloud数据进行管理。

作者简介

Jesse Feiler 是一名开发者和作家。他成为苹果开发者时，苹果开发还没有现在这么流行。他的著作包括*Sams Teach Yourself Core Data for Mac and iOS in 24 Hours* (Sams Publishing, 2011)、*Sams Teach Yourself Objective-C in 24 Hours* (Sams Publishing, 2012)、*FileMaker 12 in Depth* (Que Publishing, 2012)以及*iWork for Dummies* (Wiley, 2012)。

Jesse最初写作的有关Objective-C和苹果开发框架的书籍包括*Rhapsody Developer's Guide* (Academic Press, 1997)和*Mac OS X Developer's Guide* (Morgan Kaufmann, 2001)。他的关于苹果公司的技术，如Cyberdog、OpenDoc、ODF、Bento以及Apple Guide的书籍，在苹果开发者的书架上也占有一席之地。

他开发了Minutes Machine，这是一款面向iPad的会议管理应用，他还开发了面向iPhone和iPad的Saranac River Trail应用。可以从App Store购买这些应用，更多细节可在champlainarts.com中看到。

作为土生土长的华盛顿特区人，Jesse曾在纽约居住，目前居住在纽约的普拉茨堡，在那里他担任普拉茨堡公共图书馆的董事，以及Saranac River Trail Advisory Committee的主席。

可以通过<http://northcountryconsulting.com>联系他。

致 谢

一如既往，Waterside Productions的Carole Jelen为本书的完成提供了大量的帮助和指导。在Addison-Wesley，是Trina MacDonald促成了本书从理念到出版的过程。Michael Thurston提供了诸多出色的编辑建议。出版经理Julie Nahil，在制作技术书籍这一非常复杂的过程中，让事情不断向前推进。自由职业项目经理Anna Popick和自由职业文稿编辑Carol Lallier，极大地促成了本书的成功发行。优雅的封面设计是由Chuti Prasertsith完成的。

尽管本书是在这些人和其他许多人的帮助下才顺利付梓的，但难免出现一些问题，不过任何错误都是作者自己造成的。

前言

当苹果公司发布新产品或其操作系统的新版本时，通常会重大的新闻事件，经常有人在苹果手机商店排队等待。这些发布通常存在一些规律。对于操作系统，主要的发布是在6月的苹果全球开发者大会上进行的。在某些年份，一种或全部两种操作系统的开发者预览版本在早春可用。整个夏季，若干开发者发布版本会被提供。在秋天的某个时候，关于新iPhone变得可用的谣言开始流传，而且苹果公司会发出将在一周后举行媒体活动的邀请。在接下来的新闻事件中，新版本的iOS随着新的iPhone公映。iOS的公开发布在一周后进行，随后新的iPhone也会可用。接下来(通常在下一个个月)，iPad、Mac和OS X会重复这个过程。

在过去的几年时间，这些过程一直是计划好的，但并不保证会一直重复。值得注意的重要一点是，这些产品和操作系统的通告及发布都有具体的日期。iCloud则与之大不相同。多年来，苹果公司已经建立了重要的硬件和电信支撑结构，用于驱动iCloud和其他网络操作。同许多类似基础设施的情况一样，其细节是保密的。我们之所以知道苹果公司一些数据中心的位置，是因为它们经常需要构建许可和其他公共文档及权限，但它们通常避开公众的视线。iCloud的发布并没有剪彩或交钥匙的环节，而是一个耗时若干年的过程(并且未来还将持续若干年)。

除了硬件基础设施，iCloud还有软件组件。然而，这也经历了若干年的发展历程。正如你将在本书中看到的，iCloud有一部分在操作系统用户界面中实现，其余部分通过对现有框架和API进行相对较小的修改来实现。对于开发商和消费者来说，iCloud的公开发布是操作系统及硬件的新版本发布的一部分。

总之，iCloud不是一个产品，而是一项无处不在的技术，也是苹果的公司战略。不像苹果的硬件和软件产品，iCloud没有部件编号且没有版本号。这是跨公司各个产品的一部分。

出于这个原因，编写关于iCloud的技术资料，或学习针对它而进行的开发，都是不容易的。本书最初在2012年构想，但随着逐步成型，一件事变得清晰，就是iCloud的一些最强大的部件尚未就位。与其赶出一本不完整的书，并且依靠未来某个时候发布修订版，不如等待时日。Trina MacDonald和Addison-Wesley同意推迟出版日期，从而使之包含来自于2013年6月的WWDC的信息，我非常感激他们这样做。

你将看到，本书在我所称的 iCloud Round Trip 中达到了顶峰。在最后一章，你将看到如何建立 iOS 应用和 OS X 应用，让你可以通过 iCloud 在 OS X 和 iOS 之间共享数据。具有能够实现 Round Trip 的工具，对我来说是出版本书的一次很好的时机。这是在 iCloud 世界中你能够最大限度地接近新产品发布事件的绝佳时机。

本书读者对象

本书是为那些想探索 iCloud 的开发人员编写的。因为 iCloud 在操作系统的很多区域中实现，所以你需要熟悉 Cocoa 和 Cocoa Touch 的许多部分。随着本书对 iCloud 的介绍，作者已尝试至少概述它所涉及的各种组件。这意味着对于诸如通知的话题讨论，作者站在相当高的水平上：有些人会认为“每个人都知道那些知识”，而其他人可能认为需要介绍更多的细节。

作者一直在尝试为与 iCloud 进行交互的各种 Cocoa 技术专家和新手提供一条中间道路。苹果公司在 developer.apple.com 网络上的文档中针对更多的细节提供了主要的资源，如果觉得需要它们的话，可以参考一下。如果在触及某个领域时觉得自己已经了解这个话题，可跳到 iCloud 的细节。即使对于苹果公司的工程师而言，Cocoa 的许多知识领域是他们从里到外都知道的(也可能写过)，但还有其他一些领域是他们所不熟悉的。

依据技能和知识，你应该具有关于 Cocoa 和/或 Cocoa Touch 以及 Xcode 的基本知识。为了理解代码，掌握 Objective-C 是必需的。作者所著的 *Sams Teach Yourself Objective-C in 24 Hours* 一书提供了有关该话题的介绍。

此外，你应该具有 iCloud 的使用经验。尝试针对一项并未使用过的技术进行开发的人，数量之多总是令人惊讶。没有什么能够取代实际的用户体验。

下载示例文件

每章的示例文件都可以从网站 <http://northcountryconsulting.com> 或 <http://informit.com/title/9780321889119> 下载。除了示例，你还可以在这两个网站上找到任何更新和修订。下载的有些示例包含额外的代码，比如在第 14 章“在 iCloud 中使用文件封装器”中，除了 iPhone 界面，还提供了 iPad 界面。

这些文件是按章来组织的，并且它们以代码展示作为各章的结束。因此，在下一章建立在前一章代码基础之上的情况下，需要下载前一章并且仔细学习本章，在此过程中再添加新章的代码。

iCloud 需要代码签名，所以你会在本书中看到如何设置项目以实现这一点。值得注意的是，在本书及可下载的代码中所包含的代码签名，无法在你的计算机上工作。你必须使用自己的开发者凭据，而不是让代码签名信息保持空白，我使用了自己的凭据(不提供密码，甚至开发商账户名称也已更改)。这意味着，除非定制自己的开发者账户，否则代码将不能运行。这是有意的，也是必要的。

代码主要是针对 Xcode 5 和 OS X Mavericks (10.9)编写的。

本书的组织结构

本书分为如下 4 部分：

第 I 部分：iCloud 简介

第 I 部分提供了关于 iCloud 的用户视角和开发者视角。

第 1 章，“探讨 iCloud 及其用户体验”：随着 iCloud 的演变，它已被纳入一些应用，如 iWork 套件。你会看到 iCloud 对于应用和操作系统的用户界面是什么样子。

第 2 章，“搭建 iCloud 开发环境”：本章提供了 iCloud 的 API 结构概述。这是本书其余部分的路线图。

第 II 部分：使用 API

这一部分探讨了如何使用用户输入和维护的 iCloud 数据。对于许多用户来说，iCloud 在他们的音乐存储，以及日历和联系人的同步中，发挥了一定的作用。有一些 API 允许开发者利用这些同步的用户数据，而本书的这一部分对它们进行了说明。使用 iCloud 能够让开发者得到很大的回报：苹果公司的工程师和用户已经完成了所有的工作，你所要做的，就是帮助用户以新的富有创造性的方式来使用他们自己的数据。

第 3 章，“介绍 API 和第一个应用”：iCloud 最简单的部分由管理用户数据的 API 组成。本章提供了本书中这一部分的路线图。

第 4 章，“使用 Contacts 的 AddressBook API”：AddressBook API 允许开发者访问和更新地址簿数据。本章将介绍实现这一点所需的基础知识。

第 5 章，“使用 Event Kit API 管理日历和提醒”：本章将介绍如何运用日历和提醒。

第 6 章，“保护用户数据隐私”：iCloud 带来了许多隐私问题，你需要在自己的应用中加以解决。这是用户的数据，你必须按照本章描述的规则来处理。

第 III 部分：运用技术

iCloud 集成了各种数据管理技术和设计模式。使用这些技术可能意味着你的应用可以充分利用 iCloud 同步功能。这些技术与 iCloud 相集成，但它们在 iCloud 出现之前已经存在了很长时间。集成才是新内容。

第 7 章，“介绍 block、线程和通知”：本章提供了 iCloud 背景下的技术路线图。即使知道这些技术，在 iCloud 的世界里回顾它们，也是很重要的。

第 8 章，“使用键-值编码”：键-值编码在 Cocoa 中已被运用多年。这是存储相对少量数据的一种非常有效的方式。使用 iCloud 可以使得它对于你和你的用户都很简单。

第 9 章，“在 iCloud 中使用 Preferences、Settings 和 Keychains”：Preferences(OS X) 和 Settings(iOS) 是键-值编码的一种特殊情况。本章介绍了如何能够将它们添加到应用中，使之适用于用户的所有设备。你还可以看到如何排除来自 iCloud 中的特定偏好与设置，如果它们不适合某个特定设备的话。

第 10 章，“使用 Core Data 管理持久存储”：Core Data 是 Cocoa 和 Cocoa Touch 中主要的数据持久化工具。本章提供了高层概述。在第 17 章“使用 Core Data 和 iCloud”中将进一步探讨。

第 11 章，“使用 Xcode 项目工作区进行协同开发”：在 Xcode 4 中引入的 Xcode 工作区，使得开发者很容易在项目中设置多个目标，并且在目标之间共享特定的文件。例

如,这将使你能够在 OS X / iOS Round Trip 中共享 Core Data 数据模型(模式)及其特定的托管对象类。

第 12 章,“使用包和资源向应用添加数据”:这是在应用中管理数据的最普遍方式。不直接使用 iCloud,但可能是 iCloud 应用十分合适的添加物,与 iCloud 同步数据互补。

第IV部分: 使用 iCloud 文档和数据

本书的最后一部分汇集了文档和文件封装器中的 API 及技术。你会看到如何在 OS X 以及 iOS 中实现它们。此外,你将看到如何完成一次 Round Trip,将文件在 iOS 和 OS X 之间同步。

第 13 章,“添加 iCloud 基础设施”:本章介绍 iCloud 所要使用的基础设施——用于与 iCloud 建立联系、管理 iCloud 可用性变化及修改 iCloud 账户的代码。请注意,这些代码在随后的每章中都需要实现。在本书此部分的后续章节中,为了关注具体问题,有些代码并没有重复。

第 14 章,“在 iCloud 中使用文件封装器”:文件封装器实现类似于 Finder 中的包的结构,对用户而言看似单独文件的文件集合。它们是一种能够有效利用 iCloud 同步功能的结构。

第 15 章,“使用 iOS 文档”:本章探讨了基于 UIDocument 的 iOS 文档模型。你会看到如何实时监视 iCloud 文档的变化。

第 16 章,“使用 OS X 文档”:在 OS X 中,Cocoa 为你管理 iCloud 文件的改变,所以你要做的工作比第 15 章中更少。然而,仍然有许多工作要做,本章将介绍如何使用 NSDocument 来完成必要的事情。

第 17 章,“使用 Core Data 和 iCloud”:本章为你提供了使用 iCloud 管理基于 Core Data 的应用的代码。本章建立在第 10 章的基础之上。

第 18 章,“完成 Round Trip”:最后,将介绍如何组建 Round Trip。记得将第 13 章的代码添加到你的两个目标(OS X 和 iOS)中。

引 言

组成Cocoa和Cocoa Touch的框架包含类、协议、定义的常数,以及其他一些支持性质的元素,包括动态共享库。最基本的框架是Foundation、UIKit(Cocoa Touch, 针对iOS)和AppKit(Cocoa, 针对OS X)。更专用的框架,如Core Audio Kit Framework, 由开发者按需使用。

iCloud是不同的。不要搜寻iCloud框架,因为根本不存在。甚至不要寻找iCloud API。有一些专用于iCloud的方法,但数量很少。事实上,它们相隔很远,在这个意义上,它们分散在各种不同的类和框架中。URLForPublishingUbiquitousItemAtURL:expirationDate:error:是NSFileManager类的一部分(在这个类的52个方法中,有7个与iCloud相关),而NSPersistentStoreDidImportUbiquitousContentChangesNotification是NSPersistentStoreCoordinator类的一部分(它是这个类中iCloud相关的两个通知之一)。

iCloud以这种方式实现,意味着现有的不使用iCloud的应用不受影响。此外,由于iCloud跨越多个设备以及操作系统(OS X和iOS),很难想象它能一直实现为单一的框架或API。

随着为Cocoa和Cocoa Touch API添加的这些东西,iCloud的实现依赖于一些长期最佳实践,它们现在已经被转换为精华。设计模式,例如键-值编码,可以追溯到很早的版本,并已使用了四分之一个世纪,它们在iCloud中以新的方式被使用,尽管在大多数情况下,不需要做任何事情就可以利用iCloud的功能。

Core Data一直是管理应用持久性数据的最强大解决方案,已与iCloud深度集成。然而,集成主要是(但不完全)在幕后进行的。如果不使用iCloud,使用现有的Core Data代码即可。iCloud对Core Data最显著的影响是,在过去,创建可随应用发布的数据存储时有两种方式:可以在应用的Bundle中放置种子数据库,也可以用编程的方式向应用第一次启动(或当种子数据需要被重新创建)时创建的空的数据库添加种子数据。这两种技术都已使用多年。iCloud对Core Data最大的影响是,必须使用第二种方式,第一种方式将无法正常工作。这几乎不是什么重大变化。

也许iCloud对开发者最明显的影响是,控制应用可以在运行时环境中做什么的授权机制得到了增强。授权机制实施了新的沙盒规则,开始对iCloud中的共享文档发挥作用。明确的授权和沙盒限定了应用所要使用的操作系统功能和能力,以及磁盘存储中应用能够写入数据的特定部分。它们提升了两种操作系统的稳定性和安全性。它们在iOS上必需,在OS X上可选。在这两种操作系统上,它们实施得更加积极。此外,从开发者的角度来看,你可能会

很高兴听到，Xcode 5 中授权功能的开发者界面现在发生了很大变化，得到了极大简化(沙盒与 iCloud 相关，但它们是两个独立的功能)。

iCloud 的实现经历了若干年；在 2012 年中期，OS X Mountain Lion (10.8) 和 iOS 6 的发布集成了之前一年发布的若干部件。2013 年的秋天，OS X Mavericks (10.9) 和 iOS 7 重新定义了 iCloud，并且扩展了其面向开发者的幕后工具。如果没有使用过任何的 iWork 应用 (Numbers、Pages 和 Keynote)，尝试在多个启用了 iCloud 的设备上使用其中一个。它们提供了用户视角下的 iCloud 最佳展示。

其实，这种说法是错误的。它们提供了从用户角度来看的 iCloud 最佳展示——直到开发自己的启用了 iCloud 的应用为止。

目 录

第 I 部分 iCloud 简介

第 1 章 探讨 iCloud 及其用户体验	3
1.1 云计算介绍	3
1.2 理解 iCloud 范式	4
1.2.1 通过应用组织文件	5
1.2.2 使用 iCloud、Time Machine 和 Auto Save 来管理文件	7
1.2.3 在设备之间同步数据	8
1.2.4 进行往返旅程	8
1.3 本章小结	9
1.4 练习	9
第 2 章 搭建 iCloud 开发环境	11
2.1 在 iOS 和 OS X 上管理应用安全	11
2.1.1 在 developer.apple.com 上识别你和你的应用	12
2.1.2 确定运行时的用户和泛在容器	15
2.2 深入 iCloud 基础	15
2.2.1 Apple ID	16
2.2.2 bundle 标识符	18
2.2.3 权限和功能	19
2.2.4 泛在容器	20
2.3 在应用中使用 iCloud	21
2.4 本章小结	22
2.5 练习	22

第 II 部分 使用 API

第 3 章 介绍 API 和第一个应用	25
3.1 苹果开发者入门	25
3.2 查看 API	26
3.3 介绍内置数据应用	27
3.3.1 跟上苹果公司的步伐	28

3.3.2 应用概述	29
3.3.3 为 iOS 和 OS X 创建单独的 Xcode 项目	30
3.4 装配界面	37
3.4.1 装配 iOS 界面	37
3.4.2 装配 OS X 界面	39
3.5 本章小结	40
3.6 练习	40
第 4 章 使用 Contacts 的 AddressBook API	43
4.1 考虑 iOS 和 OS X 上的 AddressBook API	43
4.2 从 iOS 应用发送邮件	44
4.2.1 确保可以发送邮件	45
4.2.2 发送消息	45
4.2.3 检查邮件已配置并且网络可用	48
4.3 从 OS X 应用发送邮件	49
4.4 使用属性列表进行存储和同步	50
4.5 本章小结	51
4.6 练习	51
第 5 章 使用 Event Kit API 管理日历和提醒	53
5.1 探讨事件类的层次结构	54
5.2 设置 OS X 的权限	54
5.3 操作 Calendar 数据库	55
5.3.1 对 Event Store 的分配和访问	55
5.3.2 创建新的事件或提醒	57
5.3.3 搜索事件或提醒	59
5.3.4 设置或修改属性	59
5.3.5 提交更改	61

5.4	向 iOS 应用添加提醒	62
5.5	在 OS X 上将事件添加到应用	64
5.6	本章小结	66
5.7	练习	66
第 6 章	保护用户数据隐私	67
6.1	隐私需求	67
6.2	查看苹果公司公布的规则和指南	67
6.3	应用隐私的最佳实践	68
6.3.1	知道什么应该是私有的	68
6.3.2	使用良好的编程风格来强制隐私	68
6.3.3	调试时要小心	69
6.3.4	要求许可并解释如何操作数据	69
6.3.5	不能要求使用个人数据来解锁应用	70
6.3.6	添加额外的措施来保护未成年人	70
6.3.7	为支持材料提供隐私	70
6.3.8	考虑用户的问题	71
6.4	本章小结	71
6.5	练习	72

第III部分 运用技术

第 7 章	介绍 block、线程和通知	75
7.1	跟上 block 和线程	76
7.1.1	队列和线程	76
7.1.2	block	76
7.2	使用通知, 赶上潮流	77
7.2.1	通知的属性	78
7.2.2	注册通知	78
7.2.3	发布通知	79
7.3	接收 iCloud 可用性变化的通知	79
7.4	介绍第二个项目	80
7.5	准备继续前进	80

7.6	本章小结	80
7.7	练习	81
第 8 章	使用键-值编码	83
8.1	建立受控的测试环境	83
8.2	实现 KVC	84
8.3	在 iOS 模拟器上测试 iCloud	85
8.4	准备测试项目	85
8.5	为 Round Trip 共享键-值存储	87
8.6	设置和使用 NSUbiquitousKey-ValueStore	88
8.6.1	查看方法	88
8.6.2	使用存储	89
8.6.3	准备用户界面	89
8.6.4	在运行时设置存储	90
8.6.5	监控存储变化	92
8.6.6	监控界面变化	93
8.7	本章小结	94
8.8	练习	95
第 9 章	在 iCloud 中使用 Preferences、Settings 和 Keychains	97
9.1	使用属性列表	98
9.1.1	查看属性列表	98
9.1.2	查看属性列表的内容	100
9.1.3	读写属性列表	101
9.1.4	在属性列表中使用 NSData 对象	102
9.1.5	在属性列表中使用标量	102
9.2	处理用户默认值	103
9.2.1	用户可以设置默认值吗?	103
9.2.2	默认值的变化有多频繁?	103
9.2.3	默认值和设置应该放在何处?	104
9.2.4	如何将用户默认值与 iCloud 一同使用?	104
9.3	注册默认值	104
9.4	本章小结	105
9.5	练习	105

第 10 章 使用 Core Data 管理持久

存储 107

10.1 了解 Core Data 的目标 107

10.1.1 理解对象图 108

10.1.2 介绍错误 108

10.1.3 介绍数据模型 109

10.2 结构化数据 109

10.2.1 属性 109

10.2.2 关系 109

10.2.3 规范化数据 111

10.2.4 反规范化数据 111

10.3 理解 Core Data 如何与 iCloud 交互 112

10.4 介绍 Core Data 项目 112

10.5 使用 Xcode 数据建模工具 114

10.5.1 管理数据模型 115

10.5.2 使用实体 116

10.5.3 将实体转换为对象 118

10.5.4 使用对象 122

10.6 探讨 Core Data 堆栈 122

10.7 本章小结 123

10.8 练习 123

第 11 章 使用 Xcode 项目工作区进行

协同开发 125

11.1 基于数字总线而构建 126

11.2 回顾 Xcode 文件管理 126

11.3 建立多项目工作区 129

11.4 创建多项目工作区 130

11.5 本章小结 133

11.6 练习 133

第 12 章 使用包和资源向应用添加

数据 135

12.1 package、bundle 和 resource 135

12.1.1 将文件添加到应用的
bundle 13612.1.2 获取 bundle 中的
文件 138

12.2 沙盒文件 139

12.2.1 建立沙盒 139

12.2.2 OS X 上的沙盒容器 141

12.2.3 写入沙盒 141

12.3 包括属性列表 142

12.3.1 将属性列表添加到
应用中 14212.3.2 将属性列表读入
NSDictionary 143

12.4 包含 Core Data 存储 143

12.5 本章小结 144

12.6 练习 144

第 IV 部分 使用 iCloud 文档和数据

第 13 章 添加 iCloud 基础设施 147

13.1 给应用添加工作区 148

13.1.1 探索 iOS 和 OS X 文档
架构的区别 150

13.1.2 处理 UI 区别 150

13.1.3 设计共享应用的文件夹
结构 150

13.1.4 检查最后的结果 151

13.1.5 审视项目 152

13.2 利用 developer.icloud.com
调试 iCloud 应用 153

13.3 创建应用 156

13.4 创建 Shared 文件夹 157

13.4.1 Constants.h 157

13.4.2 Constants.m 157

13.4.3 SharediCloudController.h 158

13.4.4 SharediCloudController.m 159

13.5 创建应用的类 168

13.5.1 AppDelegate 168

13.5.2 MasterViewController 170

13.5.3 DetailViewController 176

13.5.4 ReportDocument 179

13.6 本章小结 181

13.7 练习 181

第 14 章 在 iCloud 中使用文件

封装器 183

14.1 探索文件、文件封装器和
文档 183

14.1.1 文件 184

14.1.2	文件封装器	184	15.5	练习	252
14.1.3	文档	185	第 16 章	使用 OS X 文档	253
14.2	用户如何管理 iCloud 文件	185	16.1	NSDocument 与 UIDocument 之间区别的演化	253
14.3	开始 Placid 项目	187	16.2	规划项目	254
14.3.1	developer.apple.com 上的 证书、标识符、设备和 属性	188	16.3	开始 Chazy 项目	256
14.3.2	Xcode 5 中的证书、标识符、 设备和属性	189	16.3.1	在 Xcode 中创建应用	256
14.3.3	修改常规设置	190	16.3.2	将 Document 改为 WrappedDocument	257
14.3.4	设置图像	190	16.3.3	添加应用委托 (根据需要)	258
14.3.5	配置性能	191	16.4	编写代码	260
14.3.6	设置文档和通用类型 标识符	192	16.4.1	WrappedDocument	260
14.3.7	检查编译设置	193	16.4.2	WindowController	267
14.4	书写代码	194	16.5	测试应用	268
14.4.1	AppDelegate	195	16.6	本章小结	269
14.4.2	MasterViewController	197	16.7	练习	269
14.4.3	DetailViewController	206	第 17 章	使用 Core Data 和 iCloud	271
14.4.4	WrappedDocument	209	17.1	查看 iCloud Core Data 实现过程	271
14.5	使用故事板	214	17.1.1	为本章使用类扩展	271
14.6	本章小结	214	17.1.2	使用选项词典	272
14.7	练习	215	17.2	回退存储	273
第 15 章	使用 iOS 文档	217	17.3	创建并管理持久存储	273
15.1	规划应用的结构	218	17.3.1	异步创建持久存储	274
15.1.1	iPad 中导航控制器与拆分 视图控制器的选择	218	17.3.2	管理持久存储的变化	274
15.1.2	选定结构	218	17.4	管理账户的变化	275
15.2	开始 Loon 项目	219	17.5	数据库迁移	276
15.2.1	设置项目的基本信息	219	17.5.1	正确看待数据模型的 变化	276
15.2.2	设置项目功能	220	17.5.2	重新开始	276
15.2.3	建立文档	221	17.6	本章小结	278
15.2.4	添加设置	222	17.7	练习	278
15.3	编写代码	222	第 18 章	完成 Round Trip	279
15.3.1	AppDelegate	223	18.1	用户如何看待 Round Trip	279
15.3.2	MasterViewController	227	18.1.1	使用 OS X 的 Open 对话框	280
15.3.3	DetailViewController	240	18.1.2	使用 iOS 的拆分视图 控制器	282
15.3.4	WrappedDocument	244			
15.3.5	FileRepresentaiton	251			
15.4	本章小结	252			

18.1.3	在 OS X 的 System Preferences 中检查 iCloud 文件.....	283
18.1.4	在 iOS 的 Settings 中 检查 iCloud 文件	284
18.2	开发者如何看待 Round Trip...	286
18.2.1	使用 developer. icloud.com	286
18.2.2	使用 Xcode	287
18.3	配置共享的泛在容器	289
18.4	使用 iCloud 共享控制器	290
18.4.1	建立应用委托与控制器的 连接	291
18.4.2	声明 iCloud 共享 控制器	291
18.4.3	实现 iCloud 共享 控制器	292
18.5	将文档移至 iCloud	296
18.6	将文档从 iCloud 移至 本地存储	298
18.7	本章小结	298
18.8	练习	299